SC42030 Control for high resolution imaging
Homework Assignment 2: Optics

Ali Nawaz

August 27, 2018

This assignment is divided into four main segments. A copy of the main assignment outlining the problems is presented online[1] . Section 1 includes solutions to Geometric problems related to optics. Section 2 presents solutions to Physical optics. Section 3 outlines solutions to problems underlying Fourier methods in Physical Optics. Finally solutions to Bonus problems are worked out in Section 3. To conclude this report, supporting Matlab files are presented in Section 4.

1 Geometrical Optics

This section aims at providing solutions to some of the problems concerning the realm of fundamental geometric optics. Please refer to the questions referred to earlier. Only the solutions are provided.

1.1 55mm lens

The lens has a fixed focal length of 55mm.

1.1.1 Sensor placement distance

The objective is to figure out the sensor location if one was to take picture of a flower 5cm in diameter by placing the lens at a distance of 30cm.

This is can be solved with the aid of thin lens equation[2][pg.21] as outlined in Equation 1. The sensor should be placed at the image location, which is 67.347mm behind the lens.

$$\frac{1}{z_0} + \frac{1}{z_i} = \frac{1}{f}$$

Where, z_0 is the object distance and z_i is the image distance. And f, the focal length. (1)

$$\frac{1}{30 \cdot 10^{-2}} + \frac{1}{z_i} = \frac{1}{55 \cdot 10^{-3}}$$

$$z_i = 0.067347m = 67.347mm$$

1.1.2 Image magnification

By simple trigonometry on similarity right angled triangle, the magnification size of image can be obtained as presented with the aid of Figure 1 and expressions in Equation 2.

[1]HW 2, 2018 problem set: `https://tinyurl.com/cfhrihw2-2018`

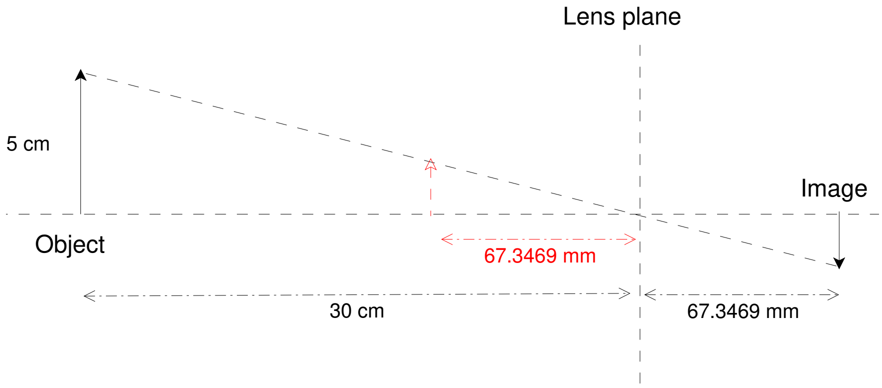


Figure 1: Figure schematising similarity triangles for magnification estimation.

$$\text{Magnification estimation by similarity triangle.} \quad \frac{\text{image height}}{67.3469 \cdot 10^{-3}} = \frac{5 \cdot 10^{-2}}{30 \cdot 10^{-2}} \qquad (2)$$
$$\text{image height} = 11.2245\ mm$$

1.1.3 Sensor placement for landscape imagery and corresponding Field Of View (FOV)

At landscape distance, the beam on the lens can be assumed to be coming from infinity. In that case, the image the image is formed at the focal length. This is simplified with the aid of Equation 3. The sensor should be placed at 55 mm behind the lens.

$$\frac{1}{z_0} + \frac{1}{z_i} = \frac{1}{f}$$
Where, z_0 is the object distance and z_i is the image distance. And f, the focal length. (3)
$$\frac{1}{\infty} + \frac{1}{z_i} = \frac{1}{f}$$
$$z_i = f = 55\ mm$$

If the sensor is placed at 55mm behind the lens and the sensor has a width of 25 mm. The FoV can be estimated with the aid of simple trigonometry as presented in Figure 2 and the expression are derived in Equation 4. Thus the FoV is 0.447 rad.

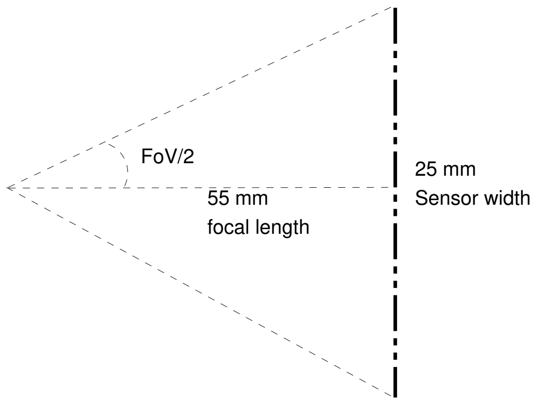


Figure 2: Figure schematising triangular geometry for FoV estimation.

$$
\begin{aligned}
\frac{\text{FoV}}{2} &= \arctan\left(\frac{(25 \cdot 10^{-3})/2}{55 \cdot 10^{-3}}\right) \\
\text{FoV} &= 0.4470\ rad
\end{aligned} \tag{4}
$$

1.2 Keplerian telescope

A Keplerian telescope as shown in Figure 3 forms the center of discussion for this problem. Distance between the lenses is equal to the sum of their focal lengths. It can be used to observe the stars with eyes, and microscopy sensor with the aid of a CCD sensor.

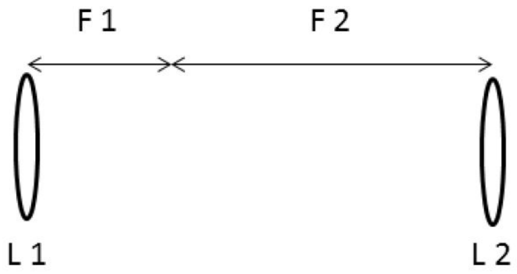


Figure 3: Overview of a Keplerian telescope.

1.2.1 Why can't we see...

Even though a Keplerian telescope can be used to observe both stars with eye and microscopy samples with a CCD sensor. One can't use use it to observe microscopy sample with naked eye and stars with a CCD sensor unless modifications are made.

To understand the reason behind it is important to understand fundamental elements of eye and a CCD sensor. A human eye, in its simplest form is a deformable lens, covered by pupil which focuses light on the retina; which in turn acts as a sensor to collect light intensity information. Where as a CCD sensor is only a sensing device. Retina is slightly curved where as CCD sensors are planar.

Figure 4 aims at demonstrating the image plane behaviour due to different object placement locations, with the aid of ray tracing. This will aid the reader in grasping the further discussions. As expected from focal point F1 till the lens plane, the image is formed at infinity. However, as the distance between the lens and the object increases beyond the focal point, the image starts to form closer and closer to the focal point of the lens.

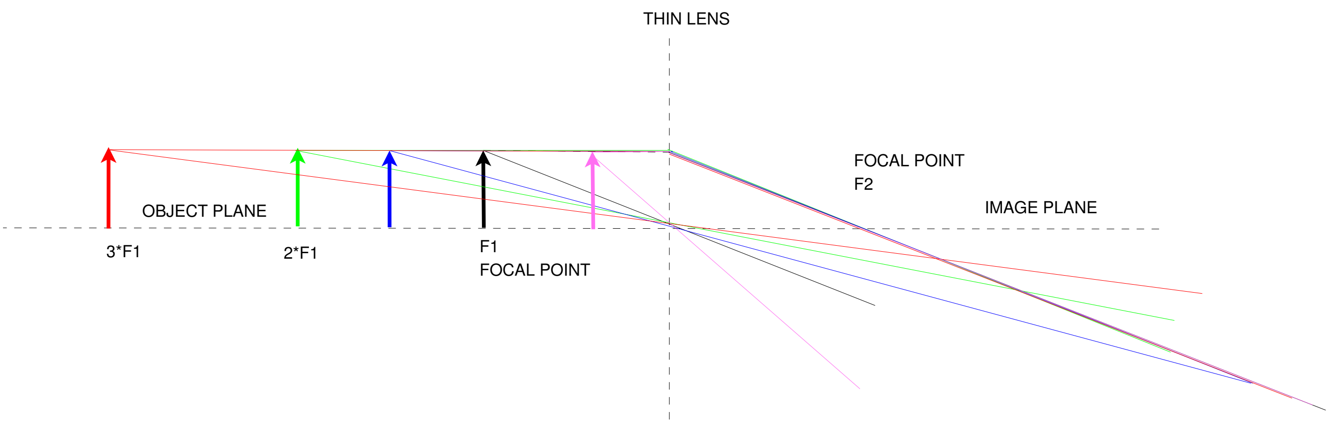


Figure 4: Ray tracing for a convex thin lens for different object placement locations.

In the case of looking at the stars the telescope is pointed at , i.e. the collimated light beams are concentrated at the focal point of the first lens behind the lens. However this image is flipped upside down. The second lens observes this flipped image at its own focal length. From thin lens equation it is known that an object placed at focal point of lens forms image at infinity behind the lens, but again flipped upside down. So the image is flipped to its correct orientation by the time it exits the second lens and an image at infinity is produced. Because the human eye has a lens before the retina, the lens facilitates focusing of the light beam on the retina. However, unlike a human eye the CCD sensor does not have a lens to focus the image on the imaging sensor plane. This can be solved by using a lens in front of CCD sensor, to concentrate the light bundle to a point onto the sensor plane.

In the case of a microscope the object is placed very close to F1, i.e. F1 + Δx. If Δx is zero or negative i.e closer to the lens side, image from the first lens is formed at infinity. If Δx is positive i.e. away from the lens side, then an image is formed closer to the lens. However, for small Δx the image can be seen a bundle of collimated beam falling on the second lens. The second lens, tries to fos the bundle of beam to a point. It is this point of image where the CCD sensor is placed to capture the image. However, the eye is not capable of sensing this, since the retina (the sensor) is not obstructed by a lens is front. This can be fixed with the addition of a movable tube which moves the distance of Lens 2. And further adding an eyepiece behind L2 which makes the bundle of light appear to be coming from infinity to the eye. The eye's lens can then concentrate the beam to the retina. This is outlined with the aid of Figure 5.

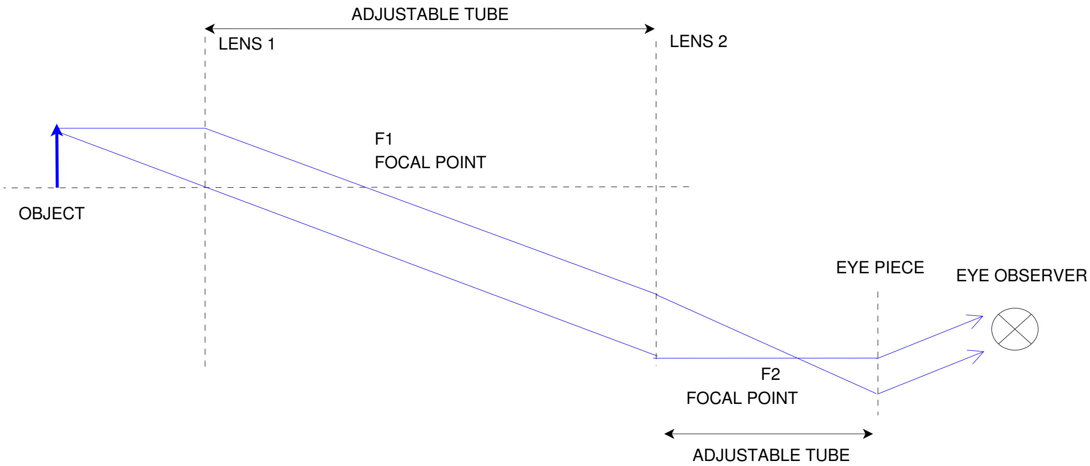


Figure 5: Adjustments made to the Keplerian telescope to make it function like a microscope.

1.2.2 Microscopy sensor placement and magnification

If the object is placed at distance f_1 from lens L_1, then the image is formed at ∞ between the two lenses. Thus the object is seen at ∞ by the second lens. The second lens thus focuses the image on the focal point F2. The CCD sensor must be place at this F2. Figure 5 can be used as an aid for visualisation.

Magnification of a thin lens is expressed with the aid of Equation 5.

$$\begin{aligned}&\text{Magnification, m} = -\frac{\text{image dimension}}{\text{object dimension}}\\&\text{-ve magnification indicates an upside down image and vice versa.}\\&\text{Magnification of two lens can be expressed as:}\\&m = m_1 \cdot m_2\end{aligned} \tag{5}$$

Thus magnification of the image observed at the CCD sensor is $\frac{-x_i}{f_1} \cdot \frac{f_2}{-x_o}$. Since image is formed by the first lens at ∞ and object observed by the second lens at ∞. The magnification of the image can be expressed as $\frac{f_2}{f_1}$.

Now if an object is place at distance $f_1 + \Delta x$, the CCD camera/sensor placement distance along with the image magnification is outlined with the aid of Equation 6.

$$
\begin{aligned}
&\frac{1}{x_o} + \frac{1}{x_i} = \frac{1}{f} \\
&\frac{1}{f_1 + \Delta x} + \frac{1}{x_i} = \frac{1}{f_1} \\
&x_i = \frac{f_1(f_1 + \Delta x)}{\Delta x} \\
&\text{Distance between image 1 and Lens 2 is the object distance for Lens 2} \\
&x'_o = f_1 + f_2 - \frac{f_1(f_1 + \Delta x)}{\Delta x} \\
&x'_o = \frac{f_2 \Delta x - f_1^2}{\Delta x} \\
&\text{The image plane of the second lens can be indicated by} \\
&\frac{1}{x'_0} + \frac{1}{x'_i} = \frac{1}{f_2} \\
&x'_i = -\frac{f_2(f_2 \Delta x - f_1^2)}{f_1^2} \\
&\text{The CCD sensor should be placed at this location } x'_i \text{ behind the second lens.} \\
&\text{Corresponding magnification of the image is given by:} \\
&m = -\frac{x_i}{x_o} \cdot -\frac{x'_i}{x'_o} \\
&m = \frac{f_1(f_1 + \Delta x)}{\Delta x(f_1 + \Delta x)} \cdot \frac{-f_2(f_2 \Delta x - f_1^2)}{f_1^2} \cdot \frac{\Delta x}{f_2 \Delta x - f_1^2} \\
&m = \frac{-f_2}{f_1}
\end{aligned}
\tag{6}
$$

1.2.3 Magnification of image for stars

Using similar analogy of multiple thin lens magnification the magnification of image for stars can be approximated. This is outlined with the aid of Equation 7.

$$
\begin{aligned}
&m_1 = -\frac{z_i}{z_o} = -\frac{f_1}{\infty} \\
&m_2 = -\frac{z'_i}{z'_o} = -\frac{\infty}{f_2} \\
&m = m_1 \cdot m_2 \\
&m = \frac{f_1}{f_2}
\end{aligned}
\tag{7}
$$

1.3 Imaging with a lens: Defocus aberration

Defocus aberration is the difference between two wavefronts.Zernike polynomial of Noll's order 4,$Z_2^0(x, y)$ can be to express wave front defocus in cartesian frame $2x^2 + 2y^2 - 1$. However, for this assignment another approach is looked into [4].The object is placed at 10 cm from a thins lens of focal length 5cm. The object is shifted to 8 cm from the lens.The objective is to find the amount of defocus added to the image.

The action of a perfect lens is to produce a spherical wave in the exit pupil. The center of curvature coincides with the center of curvation of the the x_0, y_0, z_0 frame as shown in Figure 6. The corresponding wavefront distribution(optical path difference) of an exit pupil indicates a spherical wavefront converging to a point located at distance R from the exit pupil. This is expressed with the aid of Equation 8.

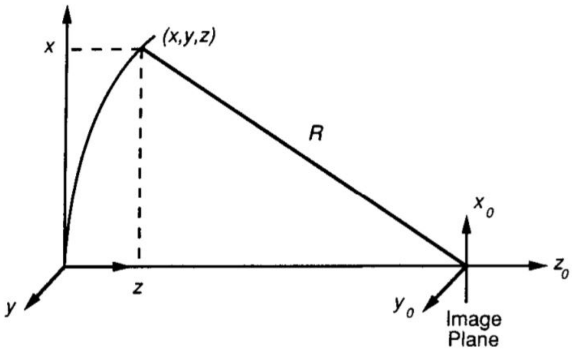


FIG. 12. Relation between image and exit pupil coordinates.

Figure 6: Spherical wavefront from a pupil exit focused at a distance R.

$$z = W(x, y) = \frac{x^2 + y^2}{2R} \tag{8}$$

For the object at 10cm from the lens of focal length 5cm, the image is formed at a distance indicated in Equation 13.

$$\begin{aligned}
&\frac{1}{z_o} + \frac{1}{z_i} = \frac{1}{f} \\
&\frac{1}{10cm} + \frac{1}{z_i} = \frac{1}{5cm} \\
&\text{Thus } z_{i,1} \text{ is at a distance of 10 cm.} \\
&\text{For the object placed at 8cm, the image is formed at:} \\
&\frac{1}{8cm} + \frac{1}{z_i} = \frac{1}{5cm} \\
&\text{Thus } z_{i,1} \text{ is at a distance of } 13\frac{1}{3} \text{ cm.} \\
&\text{Thus the difference in image plane and observation plane, } \epsilon_z \text{ is } 3\frac{1}{3}cm
\end{aligned} \tag{9}$$

This phenomena can be visualised with the aid of Figure 7. An expression outlining the corresponding wave front is presented with the aid of Equation 13

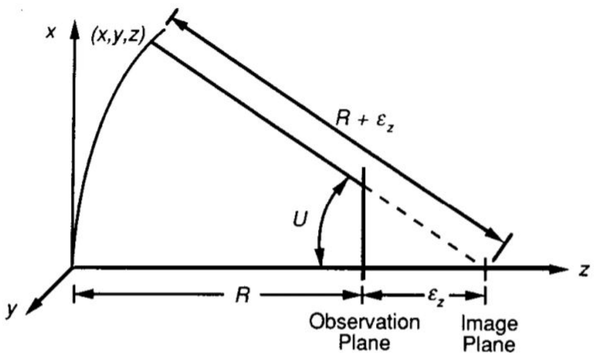


FIG. 13. Focus shift of a spherical wavefront.

Figure 7: Shift in focus of a spherical wavefront.

$$W(x,y) = \frac{x^2+y^2}{2(R+\epsilon_z)} = \frac{x^2+y^2}{2R}\left[\frac{1}{1+\frac{\epsilon_z}{R}}\right] \tag{10}$$

For ϵ_z much smaller than R, the wavefront at exit pupil can be expressed as shown in Equation 11.

$$W(x,y) = \frac{x^2+y^2}{2R} - \epsilon_z\frac{x^2+y^2}{2R^2} \tag{11}$$

The amount of wavefront defocus aberration that should be added due to OPD (Optical Path Difference) is derived in Equation 12.

$$\begin{aligned}
\Delta W(x,y) &= \frac{x^2+y^2}{2R} - \epsilon_z\frac{x^2+y^2}{2R^2} - \frac{x^2+y^2}{2R} \\
\Delta W(x,y) &= -\epsilon_z\frac{x^2+y^2}{2R^2} \\
\Delta W(x,y) &= -3\frac{1}{3}cm\frac{x^2+y^2}{2(10cm)^2} = -1\frac{2}{3}(x^2+y^2)m
\end{aligned} \tag{12}$$

2 Physical Optics

This section aims at providing solutions to some of the problems outlined under the regime of Physical optics.

2.1 Wavefront in the Keplerian telescope

The aims of this section is to present the wavefront behaviour before, inside and outside the Keplerian microscope and telescope.

Figure 8 aims at presenting the wavefront behaviour of a Keplerian telescope. Both lens have equal focal length. The telescope is pointed at the stars i.e. at ∞. Beyond the first lens the wavefront is shaped spherically. The wavefront converges at focal point 1. But the image is flipped upside down as presented earlier in this report. Since the first image plane lies in the focal point of the second lens, the image produced by the second lens is produced at ∞. This can be verified with the aid of thin lens equation.The image is flipped again, the double flipping results in actual orientation of the image.

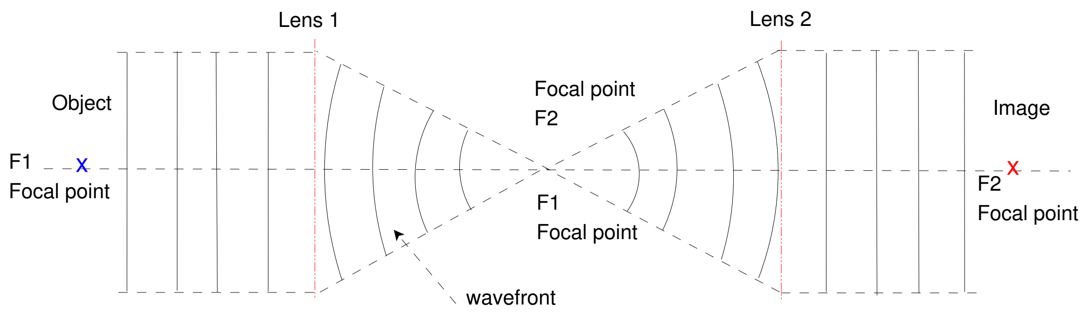


Figure 8: Wavefront behaviour in a Keplerian telescope.

Figure 9 aims at presenting the wavefront behaviour when a Keplerian telescope is used as a microscope. Both lenses are assumed to have same focal length. Assuming that the object is placed very close to the focal point of first lens, the image is produced at ∞ between both lenses. The second lens observes the object at ∞ and project an image at the focal point of second lens. Similar double flipping action upon passage through double lens occurs here. The image observed at the image point is in correct orientation.

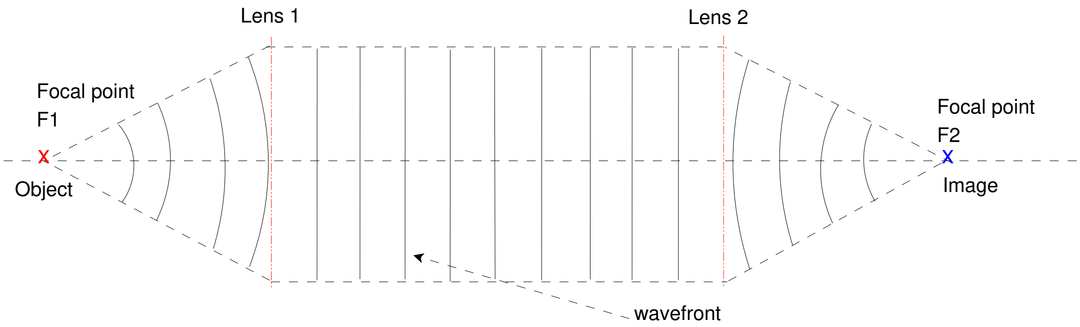


Figure 9: Wavefront behaviour in a Keplerian microscope.

Note that for images at ∞ since R is at ∞ distance, the wavefront as illustrated to be perpendicular.

2.2 Imaging with a telescope

The system is used as a microscope, to image something in the focal plane of the objective lens L_1, the objective has a focal length of 5mm with a diameter of 2 mm. The tube lens L_2 has a focal length of 20 cm and a diameter of 5 cm. The objective is to find the resolution of the system for a wavelength of 500nm. Resolution is an indication of the systems ability to resolve two object close to each other. With small angle approximation the resolution can be described with the aid of Rayleigh resolution criterion. This is indicated with the aid of Equation 13. Here, θ denotes the angular distance to the first dark ring of the airy disk. This is also known as the diffraction limit (limiting angular separation of two point sources in the sky). Since the light is captured and focused to ∞ by the first lens, L_1, its diameter of 2mm is used to estimate the angular resolution. This is the max width of the beam after lens 1. Corresponding size r (diameter) of the airy disk can be approximated with small angle trigonometric relations. This is further outlined in Equation 13. The size of the airy disk r, is an indication of the spatial resolution of the system.

$$\begin{aligned}
&\text{Angular resolution with Rayleigh criterion.}\\
&\theta = 1.22\frac{\lambda}{D}\\
&\theta = 1.22\frac{500\cdot 10^{-9}}{2\cdot 10^{-3}} = 3.05\cdot 10^{-5}\\
&\text{The size of the airy disk can be approximated by small angle approximation:}\\
&r = 1.22\frac{\lambda}{D}f_2 = 1.22\frac{500\cdot 10^{-9}}{2\cdot 10^{-3}}\cdot 20\cdot 10^{-2} = 61\cdot 10^{-6}\ [m]
\end{aligned} \tag{13}$$

Since the airy disk diameter is larger than the camera sensor pixel size of 5 μm, a better objective is recommended to achieve smaller airy disk size and hence a better resolution.

In case of a Keplerian telescope being used to watch the stars now, the angular resolution is still limited by the angular resolution of the smallest lens of the system. In this case the objective lens. This resolution will only degrade further if the observer observes the stars with a lens whose diameter is smaller than the objective lens. But at best the angular resolution of the telescope system is the angular resolution of the smallest lens used. The angular resolution is this 3.05$\cdot 10^{-5}$m as seen previously in Equation 13.

2.3 Anisoplanatic aberration in scanning microscope

A dry lens microscope is focused on a plane lying inside some specimen with uneven surface [2].Three marked regions are present on the specimen: red, green and blue. The effect of this on the image plane was introduced earlier in Figure 4. The first three arrows indicate red, green and blue in that order. As seen in the figure, the further the image object moves away the closer image plane is to the focal point of the lens. Assuming that the CCD sensor is placed at a image location where the green image plane is formed, if the the microscope is moved over the red marker, the image is formed at closer to the second lens and away from the image plane. Hence a negative aberration must be added to the image wavefront. Similarly, for the case of blue point, the image plane is now behind sensor location thus a positive aberration wavefront must be added to the image wavefront.
On the top on the image location, the surface of object also play a role in the wavefront sensed at the sensor. When the object at red is view, the observer is expected to see defocus and astigmatism aberration due to the concave surface geometry. When the green point is under observation, the observer might expect tip/tilt aberration along with coma aberration depending on the alignment of the microscope lens. Furthermore, at blue point the observer is again expected to observe defocus and astigmatism aberration due to the convex surface geometry.

2.4 Two PSFs

Two pictures of PSFs are zoomed in to shown the number of pixels present in the PSF [3]. The images are registered at $\lambda = 632nm$. Two different combinations of camera/lens are used. This is outlined as follows:

- *System 1:* Camera pixel size 5.2 μm, square aperture lens 150μm width and focal length f_1 = 3.5mm.
- *System 2:* Camera pixel size 5.5 μm, square aperture lens 300μm width and focal length f_2 = 18.2mm.

The objective is to find which camera system is used to picture the left image (picture 1, ≈10 pixels) and the right image (picture 2, ≈5 pixels). The images are assumed to be Airy disks with no aberration,i.e. for A(x) = $e^{-i\phi(x)}$, $\phi(x)$ is zero. If is further assumed that picture observed is the edges of the white circle represent locations where the intensity goes to zero and the image is observed in the focal plane. Intensity of the image filed in hte focal plane is presented with the aid of Equation 13[2][pg.28]. Here the ϕ term is zero, since absence of aberration is assumed. The outer constant term indicates the input field. P(x) defines the pupil function. For a square aperture, this is seen as the product of two rect functions.λ indicates the wavelength of the oncoming light. f the focal length. u and v are the observation plane locations used to define the angular co-ordinates of the system. While x and y represent the source plane co-ordinates. The amplitude and phase of the light at coordinates (u, v) in the focal plane are determined by the amplitude and phase of the input Fourier component at frequencies ($f_x = \frac{u}{\lambda f}$, $f_y = \frac{v}{\lambda f}$)[1]

$$I(u,v) = \frac{A^2}{\lambda^2 f^2} \left| \int\int_{\in^2} e^{-i\phi(x)} P(x) e^{-i\frac{2\pi}{\lambda f}(xu+yv)} dx \right|^2 \tag{14}$$

The expression outlined in Equation 13 can be seen as the Fourier Transform of input field × Pupil function × phase term. This is expressed with the aid of Equation 15.

[2] See Figure 2: `https://tinyurl.com/cfhrihw2-2018`
[3] See Figure 3: `https://tinyurl.com/cfhrihw2-2018`

$$\frac{A^2}{\lambda^2 f^2}\left|\mathscr{F}\left(P(x)e^{-i\frac{2\pi}{\lambda f}(xu+yv)}dx\right)\right|^2$$

The pupil function is P(τ_x,τ_y) = $\prod(\frac{x}{\tau_x})\prod(\frac{y}{\tau_y})$ and $\tau_x = \tau_y = \tau$ = lens width

For zero intensity point on the airy disk:

$$\frac{A^2}{\lambda^2 f^2}\left|\tau_x sinc(\pi\tau_x \cdot f)\tau_y sinc(\pi\tau_y \cdot f)\mathscr{F}\left(e^{-i\frac{2\pi}{\lambda f}(xu+yv)}dx\right)\right|^2 = 0$$

Note that normalized sinc function is used instead of the unnormalised sinc function.

$$sinc(\pi\tau_x \cdot f_x)sinc(\pi\tau_y \cdot f_y) = 0$$

$$\frac{sin(\pi\tau_x \cdot f_x)}{\pi\tau_x \cdot f_x}\frac{sin(\pi\tau_y \cdot f_y)}{\pi\tau_y \cdot f_y} = 0 \tag{15}$$

Since, the aperture used is a square aperture $\tau_x = \tau_y = \tau$

$$sin(\pi\tau \cdot f_x)sin(\pi\tau \cdot f_y) = 0$$

$$\sin(\pi\tau\frac{u}{\lambda f})\sin(\pi\tau\frac{v}{\lambda f}) = 0$$

$$\frac{\pi\tau u}{\lambda f} = 2\pi \textit{ and } \frac{\pi\tau v}{\lambda f} = 2\pi$$

Thus, $u, v = \frac{2\lambda f}{\tau}$

The radius of airy disk is thus $\frac{2\cdot\lambda\cdot f}{\tau}$. For system 1 and 2, the diameter of the airy disk is outlined with the aid of Equation 16.

For system 1, the radius of airy disk is:

$$r_1 = \frac{2 \cdot 632 \cdot 10^{-9} \cdot 3.5 \cdot 10^{-3}}{150 \cdot 10^{-6}} = 29.493\mu\text{m}$$

Expected number of pixels for system 1:

$$p_1 = \frac{29.493}{5.2} = 5.6717 \text{ pixels} \tag{16}$$

For system 2, the radius of airy disk is:

$$r_2 = \frac{2 \cdot 632 \cdot 10^{-9} \cdot 18.2 \cdot 10^{-3}}{300 \cdot 10^{-6}} = 76.683\mu\text{m}$$

Expected number of pixels for system 2:

$$p_2 = \frac{76.683}{5.5} = 13.9424 \text{ pixels}$$

Picture 1 has $\approx$10 pixels and and picture 2 has $\approx$ 5 pixels. Judging the results outlined with the aid of Equation 16, system 1 is used to capture picture 2 and system 2 is used to capture picture 1. The differences between observed pixels and estimated pixels might be due to the simplified assumptions undertaken (e.g. no abberation).

2.5 Resolution of a human eye

This section aims at estimating the resolution of a human eye focus at ∞ for the following cases:

- a normal eye, constricted pupil (2mm)
- a normal eye, dilated pupil (7mm)
- a myopic eye (-5 dioptres), constricted pupil (2mm)
- a myopic eye (-5 dioptres), dilated pupil (7mm)

Human eye can observe in the visibile spectrum, where the wavelength is defined in the range 390 to 700 nm. An average wavelength of 545 nm is used for the following calculations. A simplified model of the human eye consisting of a pupil, lens and retina is used [4].The human eye consists of a lens with focal distance of 24mm, pupil of diameter 2-7 mm and the retina is located at 24mm behind the lens.

2.5.1 Normal eye, constricted pupil 2mm

Angular resolution for normal eye constricted pupil in line with Rayleigh resolution criterion is presented in Equation 19.

$$\begin{aligned} \theta &= 1.22\frac{\lambda}{d} \\ \theta &= 1.22\frac{545 \cdot 10^{-9}}{2 \cdot 10^{-3}} = 3.3245 \cdot 10^{-4} \end{aligned} \tag{17}$$

Corresponding spatial resolution is presented with the expressions in Equation 18

$$\begin{aligned} r &= 1.22\frac{\lambda}{d} \cdot f \\ r &= 1.22\frac{545 \cdot 10^{-9}}{2 \cdot 10^{-3}} \cdot 24 \cdot 10^{-3} \\ r &= 7.9788 \cdot 10^{-6} \end{aligned} \tag{18}$$

2.5.2 Normal eye, dilated pupil 7mm

Angular resolution for normal eye dilated pupil in line with Rayleigh resolution criterion is presented in Equation 19.

$$\begin{aligned} \theta &= 1.22\frac{\lambda}{d} \\ \theta &= 1.22\frac{545 \cdot 10^{-9}}{7 \cdot 10^{-3}} = 9.4986 \cdot 10^{-5} \end{aligned} \tag{19}$$

Corresponding spatial resolution is presented with the expressions in Equation 20

$$\begin{aligned} r &= 1.22\frac{\lambda}{d} \cdot f \\ r &= 1.22\frac{545 \cdot 10^{-9}}{7 \cdot 10^{-3}} \cdot 24 \cdot 10^{-3} \\ r &= 2.2797 \cdot 10^{-6} \end{aligned} \tag{20}$$

2.5.3 A myopic eye (-5 dioptres), constricted pupil (2mm)

Dioptre is the focusing power of the normal eye. As shown in Equation 22.

[4] See Figure 4: `https://tinyurl.com/cfhrihw2-2018`

$$
\begin{aligned}
&D = \frac{1}{f}\\
&\text{For a myopic eye this changes to :}\\
&\bar{D} = D - 5 = \frac{1}{24 \cdot 10^{-3}} - 5 = 36\frac{2}{3}\ m^{-1}\\
&\text{Corresponding focal length of the eye:}\\
&f = \frac{1}{\bar{D}} = \frac{1}{36\frac{2}{3}}\\
&f = 0.0273\\
&\text{Corresponding angular resolution stays the same as for 2mm pupil case:}\\
&\theta = 1.22\frac{545 \cdot 10^{-9}}{2 \cdot 10^{-3}} = 3.3245 \cdot 10^{-4}\\
&\text{However, the spatial resolution changes.}\\
&r = 1.22\frac{\lambda}{d} \cdot f\\
&r = 3.3245 \cdot 10^{-4} \cdot 0.0273 = 9.0759\ \mu m
\end{aligned} \tag{21}
$$

2.5.4 A myopic eye (-5 dioptres), constricted pupil (7mm)

Dioptre is the focusing power of the normal eye. As shown in Equation 22.

$$
\begin{aligned}
&D = \frac{1}{f}\\
&\text{For a myopic eye this changes to :}\\
&\bar{D} = D - 5 = \frac{1}{24 \cdot 10^{-3}} - 5 = 36\frac{2}{3}\ m^{-1}\\
&\text{Corresponding focal length of the eye:}\\
&f = \frac{1}{\bar{D}} = \frac{1}{36\frac{2}{3}}\\
&f = 0.0273\\
&\text{Corresponding angular resolution stays the same as for 7mm pupil case:}\\
&\theta = 1.22\frac{545 \cdot 10^{-9}}{7 \cdot 10^{-3}} = 9.4986 \cdot 10^{-5}\\
&\text{However, the spatial resolution changes.}\\
&r = 1.22\frac{\lambda}{d} \cdot f\\
&r = 9.4986 \cdot 10^{-5} \cdot 0.0273 = 2.5931\ \mu m
\end{aligned} \tag{22}
$$

3 Fourier methods in Physical Optics

This part of the report is devoted to numerical simulation of the imaging, Section 3.1 presents solutions to fundamental problems of Fourier Transform. Section **??** aims at simulating airy pattern. Section **??** estimates PSF obtain for certain phase distributions. Section **??** aims at simulating aberrated images.

3.1 Warming up with Fourier

Alice and Bob want to verify numerically the well known Fourier transfom pair rect(x) $\rightarrow$ sinc(πf). Lacking the resources, each function y(x) was represented by a sequence of 128 values $y_n = y(x_n)$ at points x_n, uniformly distributed over the symmetrical range R, in the limits -R/2 to R/2.Their goal is to obtain the transformations of rect(x) and sinc(πf) function.

Alice considers the Discrete Fourier Transform (DFT) to be an approximation of the Fourier integral by a discrete sum. She is correct about it. The Fourier transform of a continuous time signal x(t) and inverse Fourier transform of X(f) along with its Discrete Time alternatives are presented with the aid of Equation 23[3].

Fourier Transform of continuous time is given by:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi f t} dt$$

This is a continuous function of freq. with real and imaginary parts.

Inverse Fourier transform x(t) of a frequency spectra X(f) is given by:

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{-i2\pi f t} df$$

Similarly, the Discrete Fourier Transform is expressed as:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}, \text{ k = 1,2, ..., N}$$

While the corresponding inverse Discrete transform is expressed as:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{-i2\pi kn/N}, \text{ n= 1, 2, ..., N}$$

(23)

Bob thinks DFT is a truncated Fourier Series of a sampled function. Which is also true as seen in the expressions outlined with the aid of Equation 23. In general DFT can be used to approximate continuous time FT. However, there are two points of concern, the sampling time and the truncation point of Fourier series used in DFT. Insufficient sampling time will lead to excessive errors due to aliasing and inappropriate truncation will lead to spectral distortion due to leakage. This is will be seen later in this section.The sampled rect(x) functions of Alice and Bob are presented in Figure 10. While the corresponding DFT is presented with the aid of Figures 11 and 12.

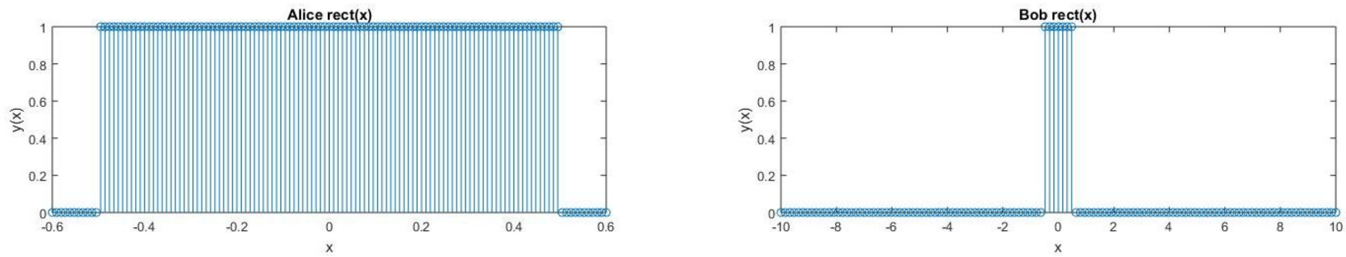


Figure 10: Sampled rect(x) signal for Alice and Bob.

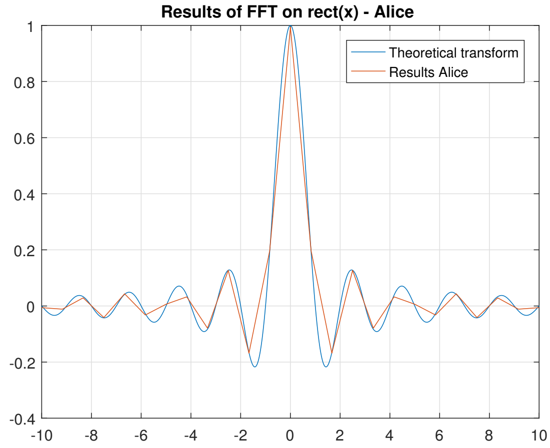


Figure 11: DFT results of Alice.

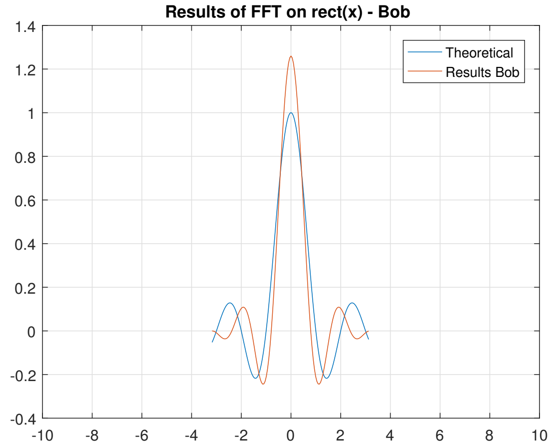


Figure 12: DFT results of Bob.

Results of Alice as seen in Figure 11 undergo spectral distortion due to leakage. This can be fixed by increasing the window width. Similarly results of Bob undergo aliasing. Since frequency components at the center and further away are not sampled fast enough due to the same amount of data points and large window width. Alice's results can be improved by increasing the sampling window width to [-2.0, 2.0]. This is visualised with the aid of Figure 13. It is important to note that increasing the sampling window width reduces the sampling frequency. Thus more data points and increased sample window is the key to good results. Figure 14 outlines an improved scenario for Bob. In this case the window width is sufficiently large, this reduces the sampling frequency. Increasing N to 600 data points while keeping the window width constant, the function-fit improves significantly.

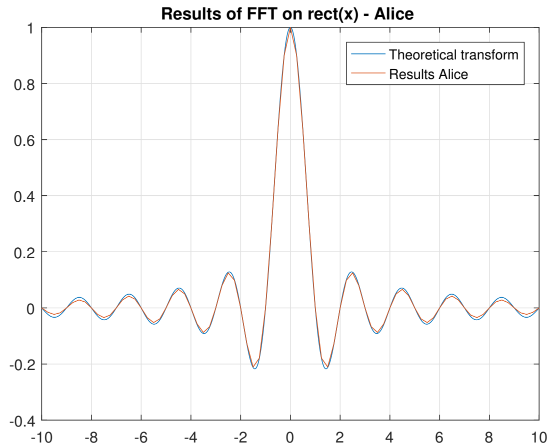


Figure 13: Improved DFT results of Alice.

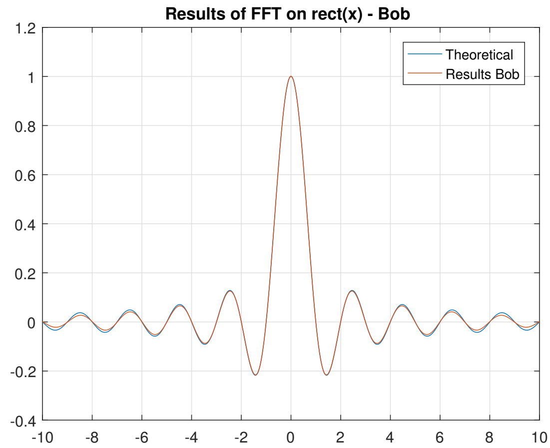


Figure 14: Improved DFT results of Bob.

If only 128 points were to be used what sampling can be used to obtain better results? This is demonstrated with the case presented in Figure13. However, an alternative is to have more sampling done at the center and decreasing sampling away from the center as the frequency of oscillations slows down away from the center and low sample points are required to capture those slow oscillations. A Matlab script outlining the working scheme is presented in Section 4.

3.2 Airy pattern Simulation

Alice and Bob now try to simulate Airy pattern by calculating the absolute value of the DFT of the single bit aperture mask. Aperture mask for Alice and Bob is presented with the aid of Figures 15 and 16. Corresponding results of PSFs computed with the aid of Matlab scripts, presented in Section 4, are presented in Figures 17 and 18.

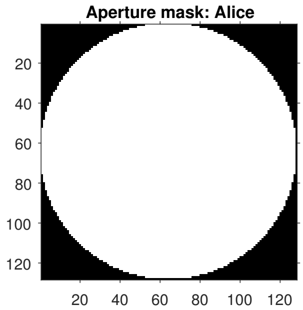


Figure 15: Aperture mask used by Alice for DFT.

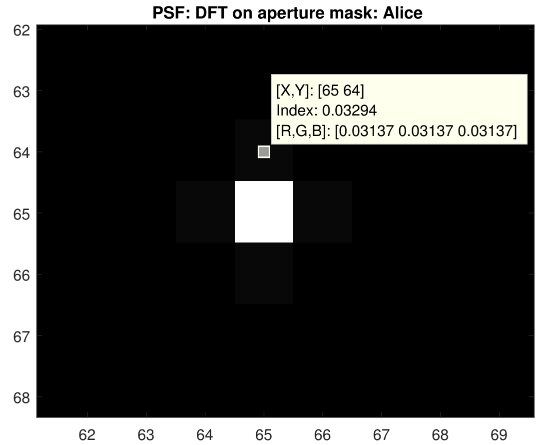


Figure 16: PSF obtained by Alice after DFT.

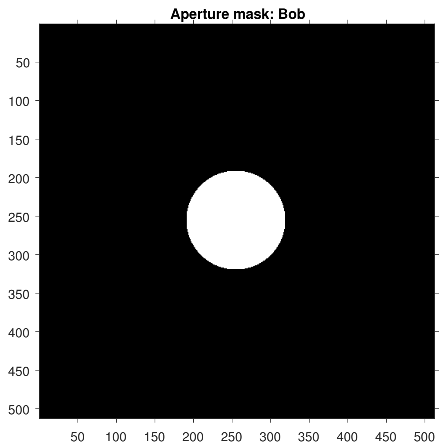


Figure 17: Aperture mask used by Alice for DFT.

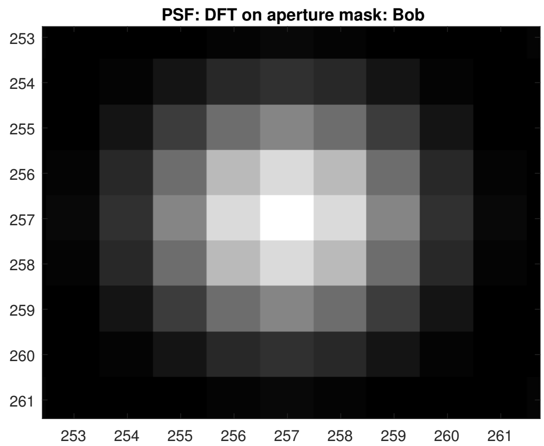


Figure 18: PSF obtained by Alice for DFT.

An important observation is made. The ratio of mask width to the aperture radius results in the image pixel width of the PSF. For the case of Alice the ratio is 128/64 thus 2 pixels should be observed. Even though Figure 15 indicates one pixel, the next two pixels are not entirely zero. Similarly for Bob 512/64 = 8 pixel width of image is observed. In line with this analogy if Alice wants to get the images in Exercise 7 [5] and still keep a circular aperture mask a similar approach can be taken. For the picture with lower resolution, where the pixel width is 5 pixels, if Alice still wants to keep the aperture diameter of 128 pixels, she should set the mask width to $5 \cdot \frac{128}{2} = 320$ pixels. Results of DFT on a mask of 320x320 pixels and an aperture of diameter 128 pixels is presented with the aid of Figures 19 and 20. Similarly if Alice wants to obtain the image with better resolution i.e. ≈ 10 pixels wide while keeping the aperture diameter of 128 pixels. She should set the mask width to $10 \cdot \frac{128}{2} = 640$ pixels. Results of DFT on a mask

[5] See Figure 3: `https://tinyurl.com/cfhrihw2-2018`

fo 640x640 pixels and an aperture of diameter 128 pixels is presented with the aid of Figures 21 and 22. A Matlab script outlining the algorithm used is presented in Section 4.

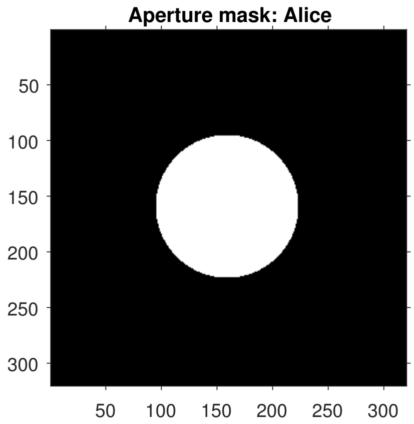


Figure 19: Aperture mask used for DFT, to generate low resolution picture.

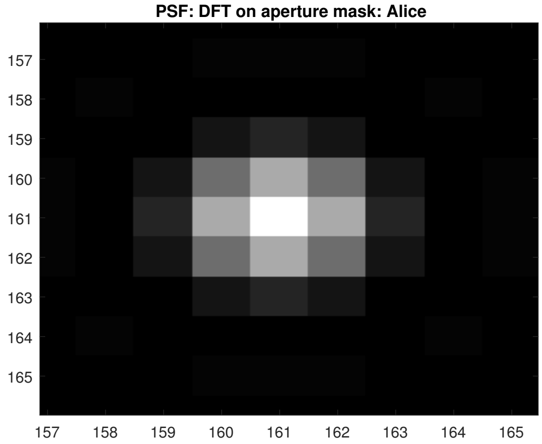


Figure 20: PSF of 5 pixels width obtained after DFT.

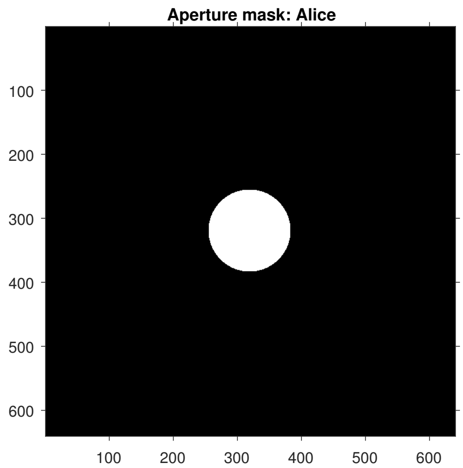


Figure 21: Aperture mask used for DFT, to generate high resolution picture.

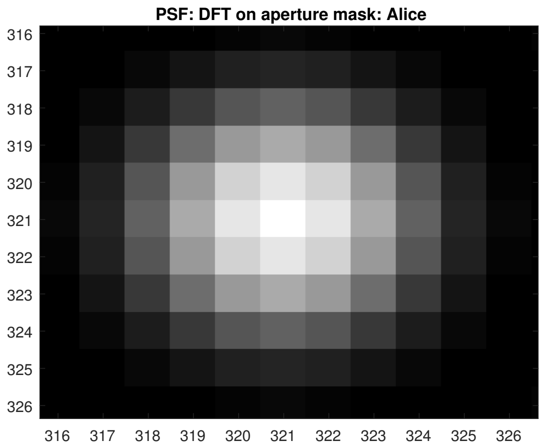


Figure 22: PSF of 10 pixels width obtained after DFT.

3.3 Aberrated PSFs

The objective of this Section is to simulate the effect of aberrated PSFs. The effect of aberration is tested on two different configuration of aperture mask outlined in the previous section. Configuration 1 is the aperture mask of dimension 320x320 $pixels^2$ with pupil diameter of 128 pixels. While configuration 2 is the aperture mask of dimension 640x640 $pixels^2$ with pupil diameter of 128 pixels. Three different configurations of aberration are tested. First one involves a scaled astigmatism: $4\pi(x^2 - y^2)$, second aberration involves a trefoil of 1μm amplitude 1μm $\cdot$(x^3 - 3xy^2). Both x and y co-ordinates are normalised with the aperture diameter. Finally two random turbulent phases in form of Kolmogorov screens, `phase1.csv` and `phase2.csv`, are presented. Script facilitating the calculations required to generate the designated images is presented in Section 4

3.3.1 Scaled astigmatism: $4\pi(x^2 - y^2)$

Figures 23 and 24 outline the aperture mask and the aberration mask used for PSF estimated for configuration 1. Figures 25 and 26 outline the aperture mask and the aberration mask used for PSF estimated for configuration 2. PSF result for configuration 1 is presented with the aid of Figure 27. PSF result for configuration 2 is presented with the aid of Figure 28.

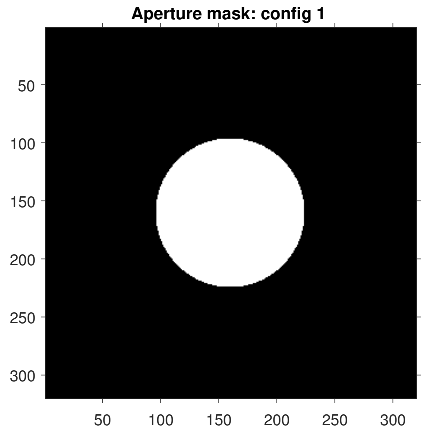


Figure 23: Aperture mask config 1.

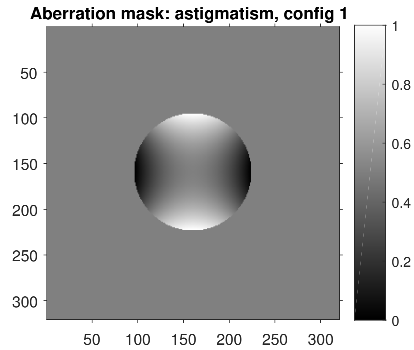


Figure 24: Aberration mask config 1.

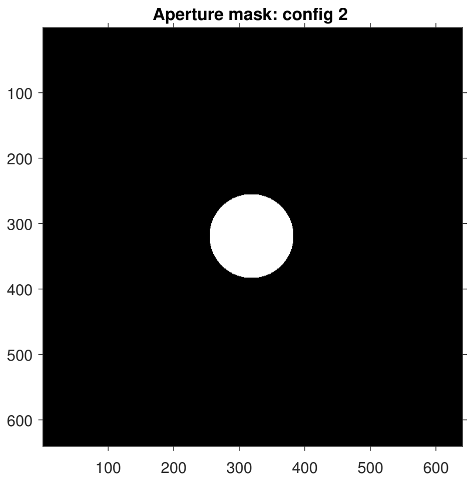


Figure 25: Aperture mask config 2.

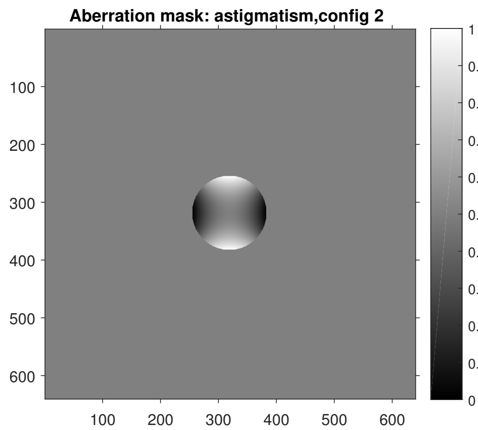


Figure 26: Aberration mask config 2.

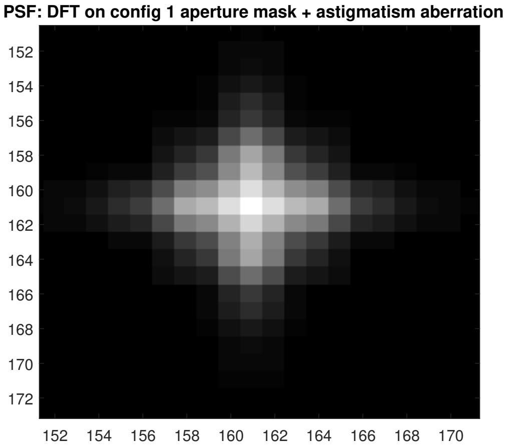


Figure 27: DFT on aperture + aberration mask config 1.

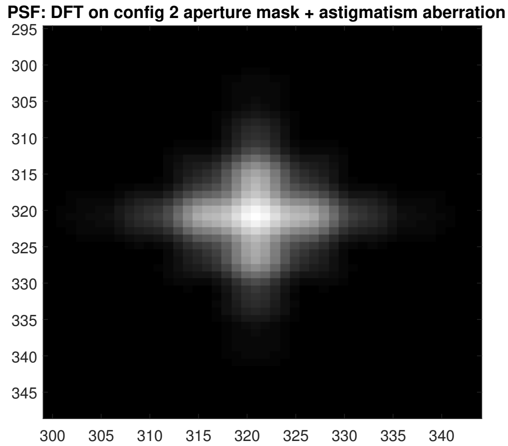


Figure 28: DFT on aperture + aberration mask config 2.

3.3.2 Trefoil of 1μm amplitude

Figures 29 and 30 outline the aperture mask and the aberration mask used for PSF estimated for configuration 1. Figures 31 and 32 outline the aperture mask and the aberration mask used for PSF estimated for configuration 2. PSF result for configuration 1 is presented with the aid of Figure 33. PSF result for configuration 2 is presented with the aid of Figure 34.

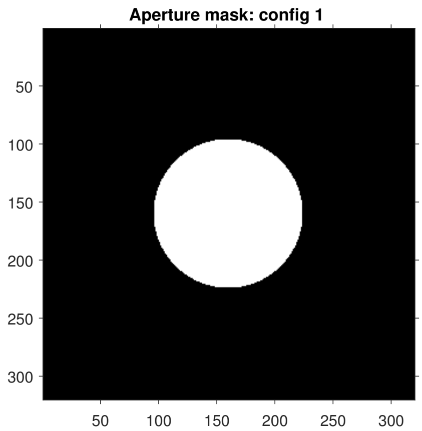


Figure 29: Aperture mask config 1.

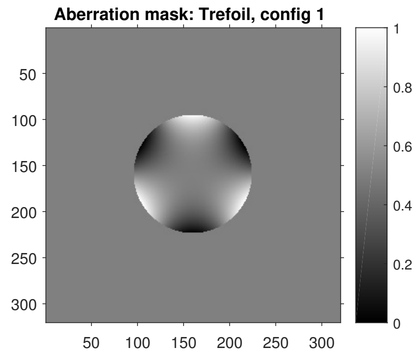


Figure 30: Aberration mask config 1.

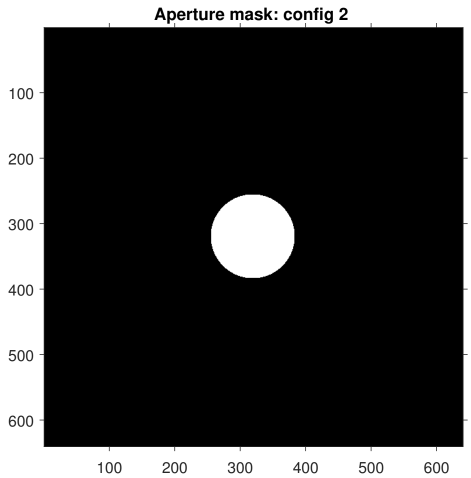


Figure 31: Aperture mask config 2.

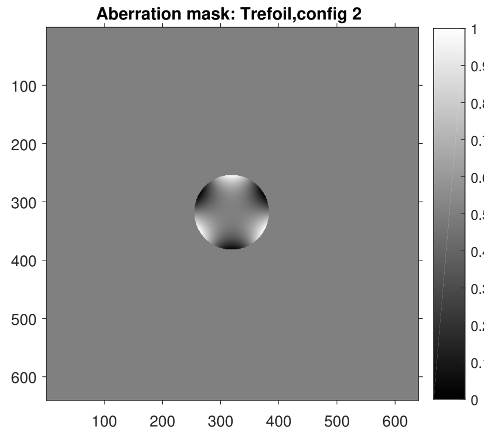


Figure 32: Aberration mask config 2.

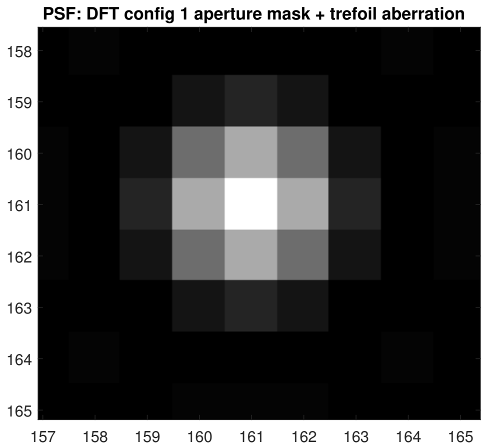


Figure 33: DFT on aperture + aberration mask config 1.

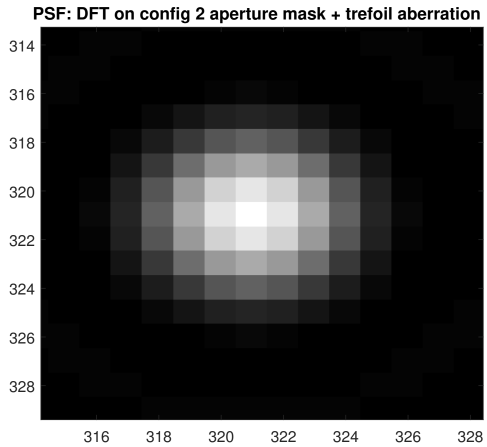
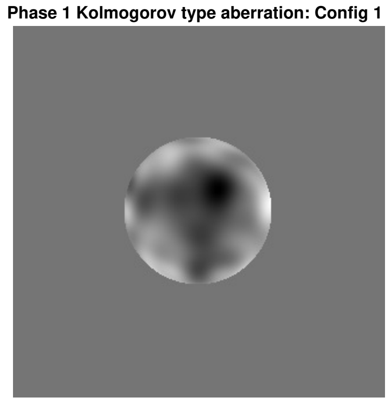


Figure 34: DFT on aperture + aberration mask config 2.

3.3.3 Kolmogorov screen phase1.csv

Figures 35 and 36 outline the phase 1 Kolmogorov type aberration mask for configuration type 1 and 2.



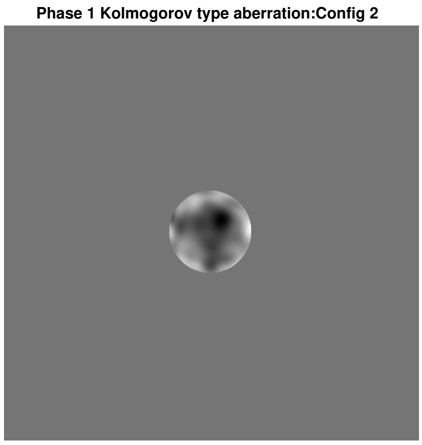
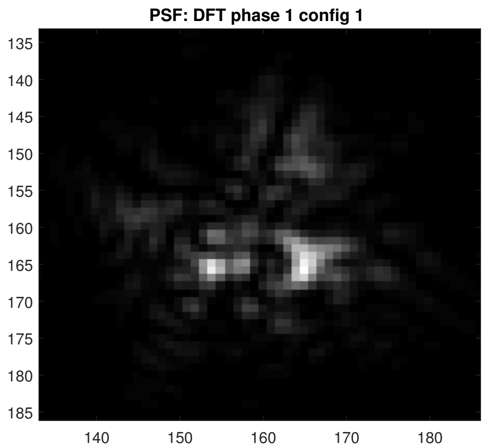


Figure 35: Phase 1 Kolmogorov aberration mask on configuration 1.

Figure 36: Phase 1 Kolmogorov aberration mask on configuration 2.

PSF for phase 1 and configuration 1 is presented in Figure37. While PSF for phase 1 and configuration 2 is presented in Figure 38.



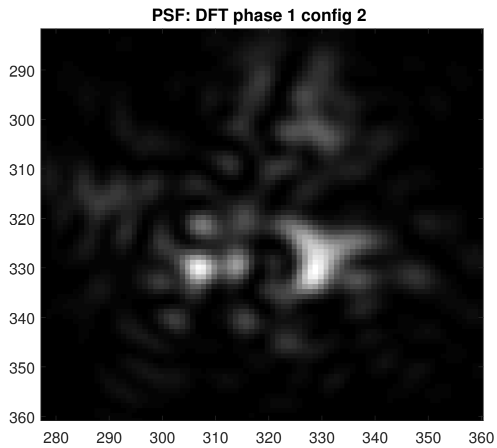
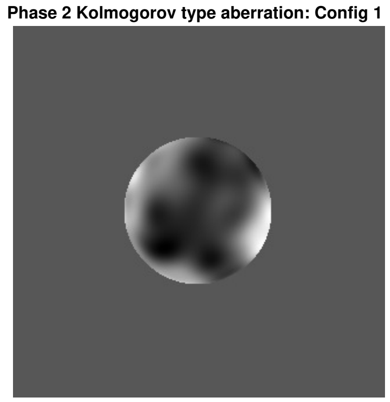


Figure 37: DFT for Phase 1 Kolmogorov aberration mask on configuration 1.

Figure 38: DFT for Phase 1 Kolmogorov aberration mask on configuration 2.

3.3.4 Kolmogorov screen phase2.csv

Figures 39 and 40 outline the phase 2 Kolmogorov type aberration mask for configuration type 1 and 2.



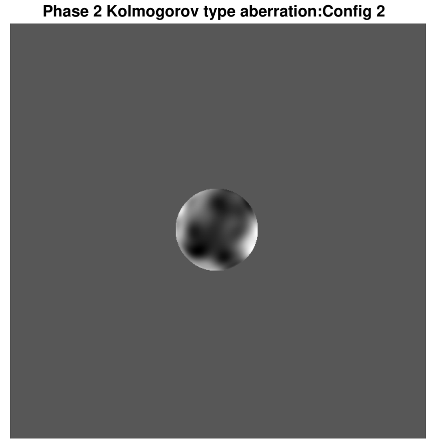
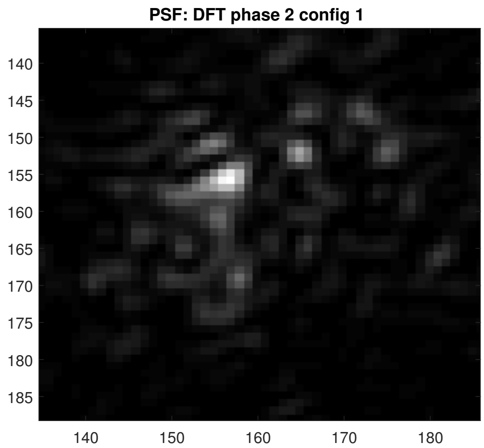


Figure 39: Phase 2 Kolmogorov aberration mask on configuration 1.

Figure 40: Phase 2 Kolmogorov aberration mask on configuration 2.

PSF for phase 2 and configuration 1 is presented in Figure41. While PSF for phase 2 and configuration 2 is presented in Figure 42.



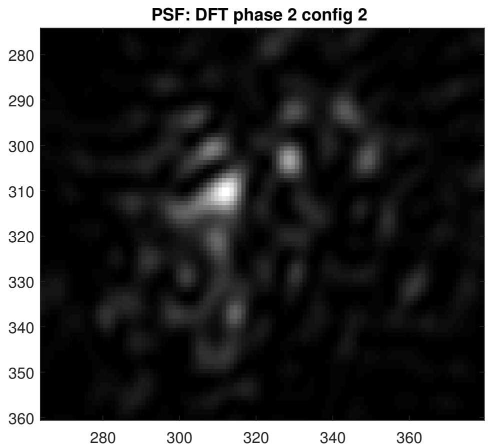


Figure 41: DFT for Phase 2 Kolmogorov aberration mask on configuration 1.

Figure 42: DFT for Phase 2 Kolmogorov aberration mask on configuration 2.

3.4 Aberrated images

This section is dedicated to simulating the diffraction limited images of 'Lena.csv' obtained with the aberrations presented in previous section i.e. scaled astigmatism, scaled trefoil and Kolmogorov turbulent screen phase 1 and phase 2. Since the aperture masks and aberration mask are presented already in the previous section, only the final images are presented here. For an ideal incoherent imaging system the image can be presented as a convolution of the object density distribution $o(\epsilon)$ [2] [pg.28] with the PSF p. The PSFs are estimated in the previous section. Thus the below results are convolution of the diffraction limited image 'Lena.csv' with designated PSFs. Matlab script used to generate the effect of 'Lena.csv' is presented along with the script of Exercise 11, in Section 4.

3.4.1 Scaled astigmatism: $4\pi(x^2 - y^2)$

Figures 43 and 44 present the effect of scaled astigmatism with different aperture configurations on Lena.

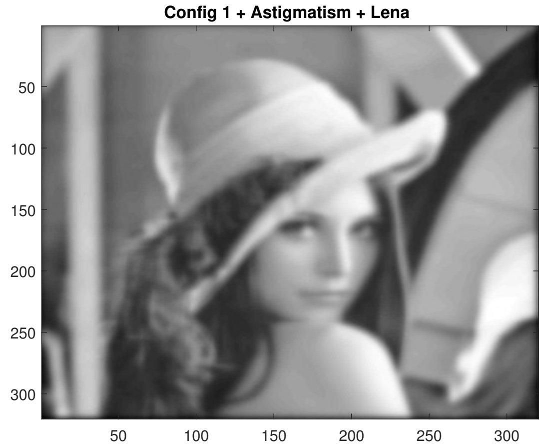


Figure 43: Lena convoluted with configuration 1 aperture and astigmatism aberration.

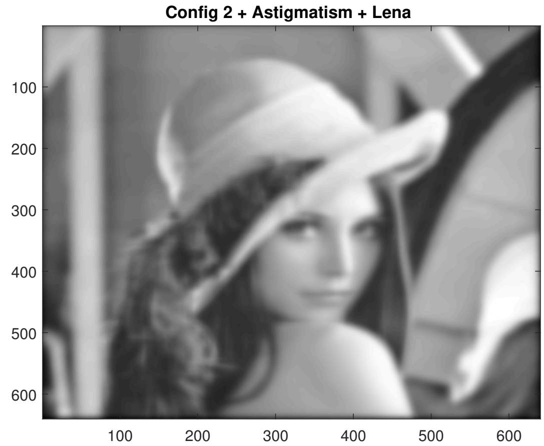


Figure 44: Lena convoluted with configuration 2 aperture and astigmatism aberration.

3.4.2 Trefoil of 1μm amplitude

Figures 45 and 46 present the effect of scaled trefoil aberration with different aperture configurations on Lena.

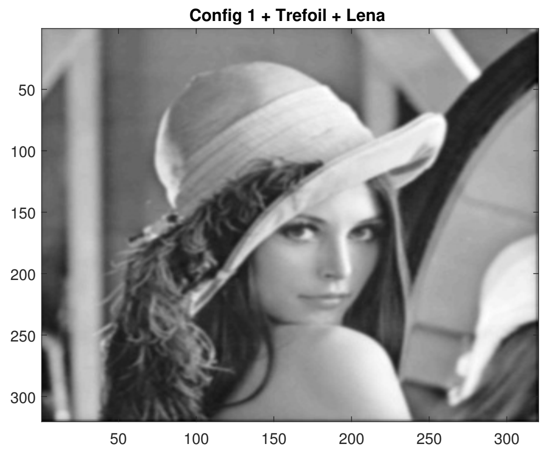


Figure 45: Lena convoluted with configuration 1 aperture and trefoil aberration.

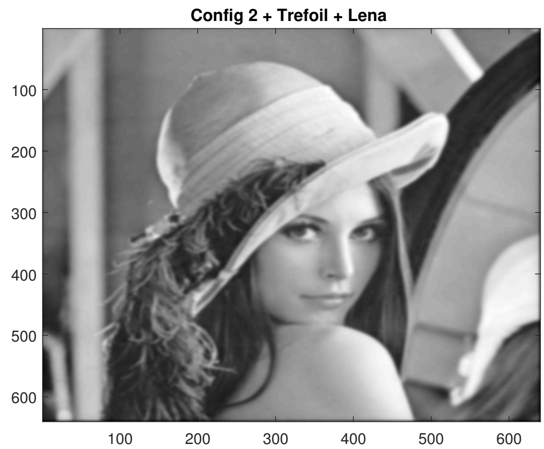


Figure 46: Lena convoluted with configuration 2 aperture and trefoil aberration.

3.4.3 Kolmogorov screen phase1.csv

Figure 47 and 48 present the effect of Kolmogorov phase 1 aberration with different aperture configurations on Lena.

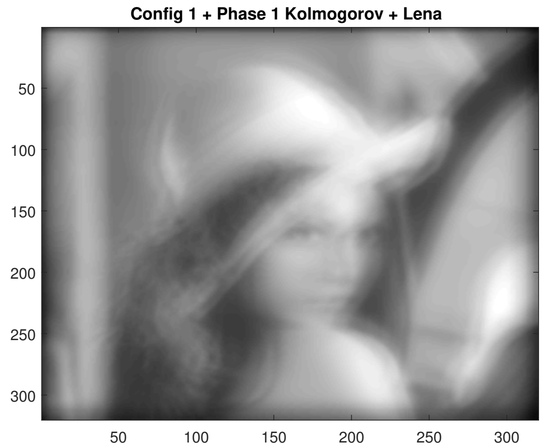


Figure 47: Lena convoluted with configuration 1 aperture and Phase 1 Kolmogorov aberration.

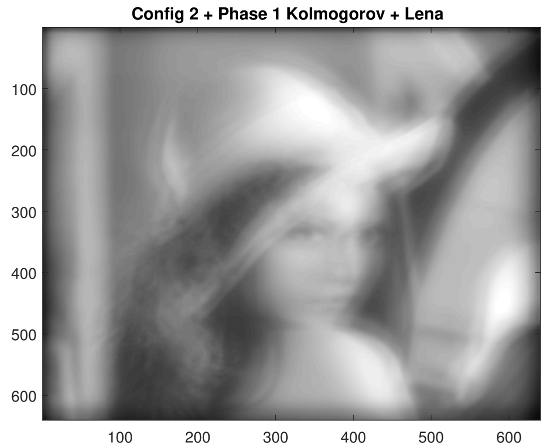


Figure 48: Lena convoluted with configuration 2 aperture and Phase 1 Kolmogorov aberration.

3.4.4 Kolmogorov screen phase2.csv

Figure 49 and 50 present the effect of Kolmogorov phase 2 with different aperture configurations on Lena.

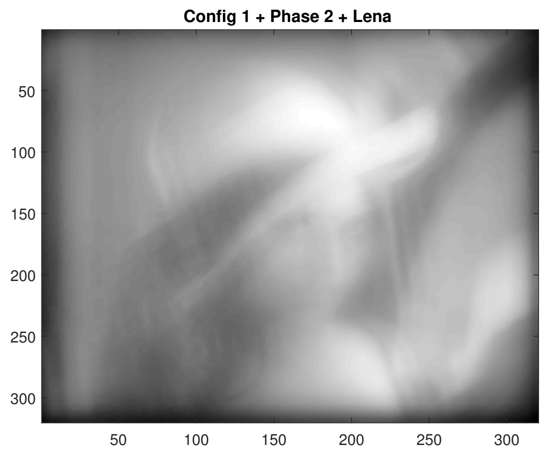


Figure 49: Lena convoluted with configuration 1 aperture and Phase 2 Kolmogorov aberration.

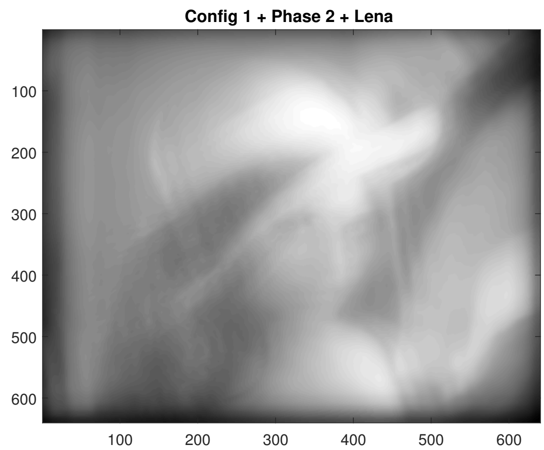


Figure 50: Lena convoluted with configuration 2 aperture and Phase 2 Kolmogorov aberration.

4 Matlab Scripts

This section aims at listing all the Matlab scripts used to generate the results.
Following is the script used to obtain the results for Exercise 9.

```
%% Exercise 9
close all;
clear all;
```

```
clc;

N = 128; % no. of sample points
% Rect signal of Alice
lb_alice = -0.6; % lower bound on window width
ub_alice = 0.6; % upper bound on window width
x_alice = [linspace(lb_alice,0,round(N/2)), linspace(0,ub_alice,round(N/2))]; % x axis

% Generate the rect signal
index_alice = find(x_alice>=-0.5 & x_alice<=0.5);
y_alice = zeros(size(x_alice));
y_alice(index_alice) = 1;

% Rect of Bob
lb_bob = -10; % lower bound on window width
ub_bob = 10; % upper bound on window width

x_bob = [linspace(lb_bob,0,round(N/2)), linspace(0,ub_bob,round(N/2))]; % x axis
% x_bob = 0.5*(lb_bob+ub_bob) + 0.5*(-lb_bob+ub_bob).*cos( (( (2.*[1:N]) - ...
    1)./(2*N))*pi); % Chebyshev polynomial data point distribution test

% Generate the rect signal
index_bob = find(x_bob>=-0.5 & x_bob<=0.5);
y_bob = zeros(size(x_bob));
y_bob(index_bob) = 1;

% Generate Rect(x) plots
figure(1)
subplot(1,2,1)
stem(x_alice,y_alice)
title('Alice rect(x)');
xlabel('x');
ylabel('y(x)');
subplot(1,2,2)
stem(x_bob,y_bob)
title('Bob rect(x)');
xlabel('x');
ylabel('y(x)');

%% DFT for Alice
R_alice = abs(lb_alice)+ abs(ub_alice); % window width
fs_alice = length(x_alice)/R_alice; % Sampling frequency
fn_alice = fs_alice/2; % nyquist frequency
fsi_alice = fs_alice/N ; % frequency of the sampling interval,

x_freq_alice = [0:fsi_alice:fs_alice-fsi_alice]; % x axis frequency points for Fourier ...
    Spectrum
y_fft_alice = fftshift(fft(ifftshift(y_alice))); % Centering around zero frequency and ...
    generating the fft of y
y_fft_alice = y_fft_alice*(1/fs_alice); % normalization with respect to the sampling ...
    frequency.

figure(2)
range_xf_alice = x_freq_alice - fn_alice; % frequency x axis for DFT plot
xf_theory_alice = [min(range_xf_alice):0.001:max(range_xf_alice)]; % High sampled x ...
    axis for Fourier transform
plot(xf_theory_alice,sinc(xf_theory_alice),x_freq_alice - fn_alice,real(y_fft_alice)); ...
    % Plotting actual results and results from Alice
grid on
legend('Theoretical transform','Results Alice')
title('Results of FFT on rect(x) - Alice')
xlim([-10 10]);

%% DFT for Bob
R_bob = abs(lb_bob)+ abs(ub_bob); % window width
fs_bob = (length(x_bob)-1)/R_bob; % Sampling frequency
fn_bob = fs_bob/2; % nyquist frequency
fsi_bob = fs_bob/N ; % frequency of the sampling interval,

x_freq_bob = [0:fsi_bob:fs_bob-fsi_bob]; % x axis frequency points for Fourier Spectrum
```

```
y_fft_bob = fftshift(fft(ifftshift(y_bob))); % Centering around zero frequency and ...
    generating the fft of y
y_fft_bob = y_fft_bob*(1/fs_bob); % normalization with respect to the sampling frequency.

figure(3)
range_xf_bob = x_freq_bob - fn_bob; % frequency x axis for DFT plot
xf_theory_bob = [min(range_xf_bob):0.001:max(range_xf_bob)]; % High sampled x axis for ...
    Fourier transform
plot(xf_theory_bob,sinc(xf_theory_bob),x_freq_bob - fn_bob,real(y_fft_bob)); % ...
    Plotting actual results and results from Alice
grid on
legend('Theoretical','Results Bob')
title('Results of FFT on rect(x) - Bob')
xlim([-10 10]);
```

Following is the script used to generate the results for Exercise 10.1.

```
% Exercise 10.1: Airy pattern simulation
clear all
close all
clc

%% Alice
nxa = 128; % number of pixels in the x-axis
nya = 128; % number of pixels in the y-axis

aperture_alice = zeros(nxa,nya); % aperture mask

x0a = nxa/2; % center of aperture
y0a = nya/2; % center of aperture
Ra = nxa/2; % radius of aperture
for z = 1:nxa
    for zz = 1:nya
        if (z-x0a)^2 +(zz - y0a)^2 < Ra^2
            aperture_alice(z,zz) = 1; % Setting value within the radius to 1
        end
    end
end

% Plotting the aperture
figure(15)
aperture_image_a = mat2gray(aperture_alice);
xylimit = imref2d(size(aperture_image_a));
% xylimit.XWorldLimits = [1 128];
% xylimit.YWorldLimits = [1 128];
imshow(aperture_image_a,xylimit)
title('Aperture mask: Alice')

%% FFT i.e. DFT on the aperture
Y = fft2(aperture_alice);
Y = abs(fftshift(Y)).^2;
figure(16)
imagesc(Y/max(max(Y)))
colormap(gray)
title('PSF: DFT on aperture mask: Alice');

%% Bob
nxb = 512; % number of pixels in the x-axis
nyb = 512; % number of pixels in the y-axis

aperture_bob = zeros(nxb,nyb); % aperture mask

x0b = nxb/2; % center of aperture
y0b = nyb/2; % center of aperture
Rb = 128/2; % radius of aperture

for z = 1:nxb
    for zz = 1:nyb
```

```
        if (z-x0b)^2 +(zz - y0b)^2 < Rb^2
            aperture_bob(z,zz) = 1; % Setting the radius to 1
        end
    end
end

% Plotting the aperture
figure(17)
aperture_image_b = mat2gray(aperture_bob);
xylimit = imref2d(size(aperture_image_b));
% xylimit.XWorldLimits = [1 512];
% xylimit.YWorldLimits = [1 512];
imshow(aperture_image_b,xylimit)
title('Aperture mask: Bob')

%% FFT i.e. DFT on the aperture
Y = fft2(aperture_bob);
Y = abs(fftshift(Y)).^2;
figure(18)
imagesc(Y/max(max(Y)))
colormap(gray)
title('PSF: DFT on aperture mask: Bob');
```

Following is the script used to generate the results for Exercise 10.2.

```
%%  Exercise 10.2 Airy pattern simulation for lower resolution picture
clear all
close all
clc
%% Alice
nxa = 320; % number of pixels in the x-axis
nya = 320; % number of pixels in the y-axis

aperture_alice = zeros(nxa,nya); % aperture mask

x0a = nxa/2; % center of aperture
y0a = nya/2; % center of aperture
% Ra = nxa/2; % radius of aperture
Ra = 128/2;
for z = 1:nxa
    for zz = 1:nya
        if (z-x0a)^2 +(zz - y0a)^2 < Ra^2
            aperture_alice(z,zz) = 1; % Setting the value withing the radius to 1
        end
    end
end

% Plotting the aperture
figure(19)
aperture_image_a = mat2gray(aperture_alice);
xylimit = imref2d(size(aperture_image_a));
% xylimit.XWorldLimits = [1 128];
% xylimit.YWorldLimits = [1 128];
imshow(aperture_image_a,xylimit)
title('Aperture mask: Alice')

%% FFT i.e. DFT on the aperture
Y = fft2(aperture_alice);
Y = abs(fftshift(Y)).^2;
figure(20)
imagesc(Y/max(max(Y)))
colormap(gray)
title('PSF: DFT on aperture mask: Alice');
%%  Exercise 10.2 Airy pattern simulation for higher resolution picture
%% Alice
nxa = 640; % number of pixels in the x-axis
nya = 640; % number of pixels in the y-axis

```

```
aperture_alice = zeros(nxa,nya); % aperture mask

x0a = nxa/2; % center of aperture
y0a = nya/2; % center of aperture
% Ra = nxa/2; % radius of aperture
Ra = 128/2; % radius of aperture
for z = 1:nxa
    for zz = 1:nya
        if (z-x0a)^2 +(zz - y0a)^2 < Ra^2
            aperture_alice(z,zz) = 1; % Setting  pixels within the radius to 1
        end
    end
end

% Plotting the aperture
figure(21)
aperture_image_a = mat2gray(aperture_alice);
xylimit = imref2d(size(aperture_image_a));
% xylimit.XWorldLimits = [1 128];
% xylimit.YWorldLimits = [1 128];
imshow(aperture_image_a,xylimit)
title('Aperture mask: Alice')

%% FFT i.e. DFT on the aperture
Y = fft2(aperture_alice);
Y = abs(fftshift(Y)).^2;
figure(22)
imagesc(Y/max(max(Y)))
colormap(gray)
title('PSF: DFT on aperture mask: Alice');
```

Following is the script used to obtain the results for Exercise 11 and 12:

```
%% Exercise 11, 12
close all;
clear all;
clc;

% Load Lena.csv
lena = load('Lena.csv');
lena_dim = size(lena); % dimension of Lena.csv
pw = lena_dim(2); % picture width
ph = lena_dim(1); % picture height
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Part 1: PSF for scaled astigmatism 4*pi(x^2 - y^2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Configuration 1: low resolution aperture
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nxa = 320; % number of pixels in the x-axis
nya = 320; % number of pixels in the y-axis

x0a = nxa/2; % center of aperture
y0a = nya/2; % center of aperture
Ra = 128/2; % radius of aperture

%% PSF generation
aperture = zeros(nxa,nya); % aperture mask
aberration = zeros(nxa,nya); % aberration mask
countx = 0;
for z = (-nxa/2):1:(nxa/2)
    countx = countx +1;
    county = 0;
    for zz = (-nya/2):1:(nya/2)
        county = county +1 ;
        if (z)^2 +(zz)^2 < Ra^2
            aperture(countx,county) = 1; % Setting the value withing the radius to 1
```

```
            aberration(countx, county) = 4*pi*( (z/(2*Ra))^2 - (zz/(2*Ra))^2 ); % ...
                Aberration active in the pattern
        end
    end
end

% Inclusion of abberation phase term
aberration = flipud(aberration);
phase_ab = exp(-i*(aberration)); % Phase aberration
aperture_ab = aperture.*phase_ab; % Addition of aberration on the aperture

% Plotting the aperture mask
figure(23)
aperture_image = mat2gray(aperture);
xylimit = imref2d(size(aperture_image));
% xylimit.XWorldLimits = [1 128];
% xylimit.YWorldLimits = [1 128];
imshow(aperture_image,xylimit)
title('Aperture mask: config 1')

% Plotting aperture mask with aberration
figure(24)
ab_image = mat2gray(aberration);
xylimit = imref2d(size(ab_image));
% xylimit.XWorldLimits = [1 128];
% xylimit.YWorldLimits = [1 128];
imshow(ab_image,xylimit)
title('Aberration mask: astigmatism, config 1')
colorbar


%% FFT i.e. DFT on the aperture with aberration mask
Y = fft2(aperture_ab);
Y = abs(fftshift(Y)).^2;
figure(27)
imagesc(Y/max(max(Y)))
colormap(gray)
title('PSF: DFT on config 1 aperture mask + astigmatism aberration');

%% Plotting the effect on Lena.csv
Y_dim = size(Y); % dimension of PSF image
Y_h = Y_dim(1);  % PSF image height
Y_w = Y_dim(2);  % PSF image width
% lenabuf = padarray(lena, [abs(Y_h/2 -ph/2) abs(Y_w/2 - pw/2)],0); % Resize object ...
    intensity distribution
lenabuf = imresize(lena, [Y_h Y_w]);
Y = conv2(lenabuf,Y,'same');
figure(43)
imagesc(Y/max(max(Y)))
colormap(gray)
title('Config 1 + Astigmatism + Lena');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Configuration 2: high resolution aperture
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nxa = 640; % number of pixels in the x-axis
nya = 640; % number of pixels in the y-axis

x0a = nxa/2; % center of aperture
y0a = nya/2; % center of aperture
Ra = 128/2; % radius of pupil aperture

%% PSF generation
aperture = zeros(nxa,nya); % aperture mask
aberration = zeros(nxa,nya); % aberration mask
countx = 0;
for z = (-nxa/2):1:(nxa/2)
    countx = countx +1;
    county = 0;
    for zz = (-nya/2):1:(nya/2)
```

```
        county = county +1 ;
        if (z)^2 +(zz)^2 < Ra^2
            aperture(countx,county) = 1; % Setting the value withing the radius to 1
            aberration(countx, county) = 4*pi*( (z/(2*Ra))^2 - (zz/(2*Ra))^2 ); % ... Aberration active in the pattern
        end
    end
end

% Inclusion of abberation phase term
aberration = flipud(aberration);
phase_ab = exp(-i*(aberration)); % Phase aberration
aperture_ab = aperture.*phase_ab; % Addition of aberration on the aperture

% Plotting the aperture mask
figure(25)
aperture_image = mat2gray(aperture);
xylimit = imref2d(size(aperture_image));
% xylimit.XWorldLimits = [1 128];
% xylimit.YWorldLimits = [1 128];
imshow(aperture_image,xylimit)
title('Aperture mask: config 2')

% Plotting aperture mask with aberration
figure(26)
ab_image = mat2gray(aberration);
xylimit = imref2d(size(ab_image));
% xylimit.XWorldLimits = [1 128];
% xylimit.YWorldLimits = [1 128];
imshow(ab_image,xylimit)
title('Aberration mask: astigmatism,config 2')
colorbar


%% FFT i.e. DFT on the aperture with aberration mask
Y = fft2(aperture_ab);
Y = abs(fftshift(Y)).^2;
figure(28)
imagesc(Y/max(max(Y)))
colormap(gray)
title('PSF: DFT on config 2 aperture mask + astigmatism aberration');

%% Plotting the effect on Lena.csv
Y_dim = size(Y); % dimension of PSF image
Y_h = Y_dim(1);  % PSF image height
Y_w = Y_dim(2);  % PSF image width
% lenabuf = padarray(lena, [(Y_h/2 -ph/2) (Y_w/2 - pw/2)],0); % Resize object ... intensity distribution
lenabuf = imresize(lena, [Y_h Y_w]);
Y = conv2(lenabuf,Y,'same');
figure(44)
imagesc(Y/max(max(Y)))
colormap(gray)
title('Config 2 + Astigmatism + Lena');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Part 2: PSF for trefoil aberration: 0.1*10^(-6)(x^3 - 3xy^2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Configuration 1: low resolution aperture
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nxa = 320; % number of pixels in the x-axis
nya = 320; % number of pixels in the y-axis

x0a = nxa/2; % center of aperture
y0a = nya/2; % center of aperture
Ra = 128/2; % radius of aperture

%% PSF generation
aperture = zeros(nxa,nya); % aperture mask
```

```
aberration = zeros(nxa,nya); % aberration mask
countx = 0;
for z = (-nxa/2):1:(nxa/2)
    countx = countx +1;
    county = 0;
    for zz = (-nya/2):1:(nya/2)
        county = county +1 ;
        if (z)^2 +(zz)^2 < Ra^2
            aperture(countx,county) = 1; % Setting the value withing the radius to 1
            aberration(countx, county) = (1*10^(-6))*((z/(2*Ra))^3 - 3*(z/(2*Ra))* ...
                (zz/(2*Ra))^2); % Aberration active in the pattern
        end
    end
end

% Inclusion of abberation phase term
aberration = flipud(aberration);
phase_ab = exp(-i*(aberration)); % Phase aberration
aperture_ab = aperture.*phase_ab; % Addition of aberration on the aperture

% Plotting the aperture mask
figure(29)
aperture_image = mat2gray(aperture);
xylimit = imref2d(size(aperture_image));
% xylimit.XWorldLimits = [1 128];
% xylimit.YWorldLimits = [1 128];
imshow(aperture_image,xylimit)
title('Aperture mask: config 1')

% Plotting aperture mask with aberration
figure(30)
ab_image = mat2gray(aberration);
xylimit = imref2d(size(ab_image));
% xylimit.XWorldLimits = [1 128];
% xylimit.YWorldLimits = [1 128];
imshow(ab_image,xylimit)
title('Aberration mask: Trefoil, config 1')
colorbar


%% FFT i.e. DFT on the aperture with aberration mask
Y = fft2(aperture_ab);
Y = abs(fftshift(Y)).^2;
figure(33)
imagesc(Y/max(max(Y)))
colormap(gray)
title('PSF: DFT config 1 aperture mask + trefoil aberration');
%% Plotting the effect on Lena.csv
Y_dim = size(Y); % dimension of PSF image
Y_h = Y_dim(1);  % PSF image height
Y_w = Y_dim(2);  % PSF image width
% lenabuf = padarray(lena, [(Y_h/2 -ph/2) (Y_w/2 - pw/2)],0); % Resize object ...
    intensity distribution
lenabuf = imresize(lena, [Y_h Y_w]);
Y = conv2(lenabuf,Y,'same');
figure(45)
imagesc(Y/max(max(Y)))
colormap(gray)
title('Config 1 + Trefoil + Lena');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Configuration 2: high resolution aperture
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nxa = 640; % number of pixels in the x-axis
nya = 640; % number of pixels in the y-axis

x0a = nxa/2; % center of aperture
y0a = nya/2; % center of aperture
Ra = 128/2; % radius of pupil aperture

```

```
%% PSF generation
aperture = zeros(nxa,nya); % aperture mask
aberration = zeros(nxa,nya); % aberration mask
countx = 0;
for z = (-nxa/2):1:(nxa/2)
    countx = countx +1;
    county = 0;
    for zz = (-nya/2):1:(nya/2)
        county = county +1 ;
        if (z)^2 +(zz)^2 < Ra^2
            aperture(countx,county) = 1; % Setting the value withing the radius to 1
            aberration(countx, county) = (1*10^(-6))*((z/(2*Ra))^3 - 3*(z/(2*Ra))* ...
                (zz/(2*Ra))^2); % Aberration active in the pattern
        end
    end
end

% Inclusion of abberation phase term
aberration = flipud(aberration);
phase_ab = exp(-i*(aberration)); % Phase aberration
aperture_ab = aperture.*phase_ab; % Addition of aberration on the aperture

% Plotting the aperture mask
figure(31)
aperture_image = mat2gray(aperture);
xylimit = imref2d(size(aperture_image));
% xylimit.XWorldLimits = [1 128];
% xylimit.YWorldLimits = [1 128];
imshow(aperture_image,xylimit)
title('Aperture mask: config 2')

% Plotting aperture mask with aberration
figure(32)
ab_image = mat2gray(aberration);
xylimit = imref2d(size(ab_image));
% xylimit.XWorldLimits = [1 128];
% xylimit.YWorldLimits = [1 128];
imshow(ab_image,xylimit)
title('Aberration mask: Trefoil,config 2')
colorbar


%% FFT i.e. DFT on the aperture with aberration mask
Y = fft2(aperture_ab);
Y = abs(fftshift(Y)).^2;
figure(34)
imagesc(Y/max(max(Y)))
colormap(gray)
title('PSF: DFT on config 2 aperture mask + trefoil aberration');
%% Plotting the effect on Lena.csv
Y_dim = size(Y); % dimension of PSF image
Y_h = Y_dim(1);  % PSF image height
Y_w = Y_dim(2);  % PSF image width
% lenabuf = padarray(lena, [(Y_h/2 -ph/2) (Y_w/2 - pw/2)],0); % Resize object ...
    intensity distribution
lenabuf = imresize(lena, [Y_h Y_w]);
Y = conv2(lenabuf,Y, 'same');
figure(46)
imagesc(Y/max(max(Y)))
colormap(gray)
title('Config 2 + Trefoil + Lena');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% PART 3.1 : PHASE 1 KOLMOGOROV TYPE ABERRATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
phase1 = load('phase1.csv');
%% Configuration 1 Aberration
nx1 = 320; % x array pixel width
ny1 = 320; % y array pixel width
r1 = 128/2; % radius of pupil in pixel width
padx1 = (nx1/2) - r1;
```

```
pady1 = (ny1/2) - r1;
phase11 = padarray(phase1,[padx1 pady1],0, 'both');
phase11_image = mat2gray(phase11);
colormap(gray)
figure(35)
imshow(phase11_image)
title('Phase 1 Kolmogorov type aberration: Config 1')
%% Configuration 2 Aberration
nx2 = 640; % x array pixel width
ny2 = 640; % y array pixel width
r2 = 128/2; % radius of pupil in pixel width
padx2 = (nx2/2) - r2;
pady2 = (ny2/2) - r2;
phase12 = padarray(phase1,[padx2 pady2],0, 'both');
phase12_image = mat2gray(phase12);
figure(36)
imshow(phase12_image)
title('Phase 1 Kolmogorov type aberration:Config 2')

%% Configuration 1 aperture generation
nxa = 320; % number of pixels in the x-axis
nya = 320; % number of pixels in the y-axis

x0a = nxa/2; % center of aperture
y0a = nya/2; % center of aperture
Ra = 128/2; % radius of aperture

aperture_1 = zeros(nxa,nya); % aperture mask
countx = 0;
for z = (-nxa/2):1:(nxa/2)
    countx = countx +1;
    county = 0;
    for zz = (-nya/2):1:(nya/2)
        county = county +1 ;
        if (z)^2 +(zz)^2 < Ra^2
            aperture_1(countx,county) = 1; % Setting the value withing the radius to 1
        end
    end
end

% Inclusion of abberation phase term
phase11 = flipud(phase11);
aperture_ab_11 = aperture_1.*exp(-i*(phase11)); % Addition of phase1 aberration on ...
    config 1 aperture

%% FFT i.e. DFT on the aperture with aberration mask
Y = fft2(aperture_ab_11);
Y = abs(fftshift(Y)).^2;
figure(37)
imagesc(Y/max(max(Y)))
colormap(gray)
title('PSF: DFT phase 1 config 1');
%% Plotting the effect on Lena.csv
Y_dim = size(Y); % dimension of PSF image
Y_h = Y_dim(1);  % PSF image height
Y_w = Y_dim(2);  % PSF image width
% lenabuf = padarray(lena, [(Y_h/2 -ph/2) (Y_w/2 - pw/2)],0); % Resize object ...
    intensity distribution
lenabuf = imresize(lena, [Y_h Y_w]);
Y = conv2(lenabuf,Y, 'same');
figure(47)
imagesc(Y/max(max(Y)))
colormap(gray)
title('Config 1 + Phase 1 Kolmogorov + Lena');
%% Configuration 2 aperture generation
nxa = 640; % number of pixels in the x-axis
nya = 640; % number of pixels in the y-axis

x0a = nxa/2; % center of aperture
y0a = nya/2; % center of aperture
```

```
Ra = 128/2; % radius of pupil aperture

aperture_2 = zeros(nxa,nya); % aperture mask
countx = 0;
for z = (-nxa/2):1:(nxa/2)
    countx = countx +1;
    county = 0;
    for zz = (-nya/2):1:(nya/2)
        county = county +1 ;
        if (z)^2 +(zz)^2 < Ra^2
            aperture_2(countx,county) = 1; % Setting the value withing the radius to 1
        end
    end
end

% Inclusion of abberation phase term
phase12 = flipud(phase12);
aperture_ab_12 = aperture_2.*exp(-i*(phase12)); % Addition of phase1 aberration on ...
    config 2 aperture
%% FFT i.e. DFT on the aperture with aberration mask
Y = fft2(aperture_ab_12);
Y = abs(fftshift(Y)).^2;
figure(38)
imagesc(Y/max(max(Y)))
colormap(gray)
title('PSF: DFT phase 1 config 2');
%% Plotting the effect on Lena.csv
Y_dim = size(Y); % dimension of PSF image
Y_h = Y_dim(1);  % PSF image height
Y_w = Y_dim(2);  % PSF image width
% lenabuf = padarray(lena, [(Y_h/2 -ph/2) (Y_w/2 - pw/2)],0); % Resize object ...
    intensity distribution
lenabuf = imresize(lena, [Y_h Y_w]);
Y = conv2(lenabuf,Y, 'same');
figure(48)
imagesc(Y/max(max(Y)))
colormap(gray)
title('Config 2 + Phase 1 Kolmogorov + Lena');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% PART 3.2 : PHASE 2 KOLMOGOROV TYPE ABERRATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
phase2 = load('phase2.csv');

%% Configuration 1 Aberration
nx1 = 320; % x array pixel width
ny1 = 320; % y array pixel width
r1 = 128/2; % radius of pupil in pixel width
padx1 = (nx1/2) - r1;
pady1 = (ny1/2) - r1;
phase21 = padarray(phase2,[padx1 pady1],0, 'both');
phase21_image = mat2gray(phase21);
colormap(gray)
figure(39)
imshow(phase21_image)
title('Phase 2 Kolmogorov type aberration: Config 1')
%% Configuration 2 Aberration
nx2 = 640; % x array pixel width
ny2 = 640; % y array pixel width
r2 = 128/2; % radius of pupil in pixel width
padx2 = (nx2/2) - r2;
pady2 = (ny2/2) - r2;
phase22 = padarray(phase2,[padx2 pady2],0, 'both');
phase22_image = mat2gray(phase22);
figure(40)
imshow(phase22_image)
title('Phase 2 Kolmogorov type aberration:Config 2')

%% Configuration 1 aperture generation
nxa = 320; % number of pixels in the x-axis
nya = 320; % number of pixels in the y-axis
```

```
x0a = nxa/2; % center of aperture
y0a = nya/2; % center of aperture
Ra = 128/2; % radius of aperture

aperture_1 = zeros(nxa,nya); % aperture mask
countx = 0;
for z = (-nxa/2):1:(nxa/2)
    countx = countx +1;
    county = 0;
    for zz = (-nya/2):1:(nya/2)
        county = county +1 ;
        if (z)^2 +(zz)^2 < Ra^2
            aperture_1(countx,county) = 1; % Setting the value withing the radius to 1
        end
    end
end

% Inclusion of abberation phase term
phase21 = flipud(phase21);
aperture_ab_21 = aperture_1.*exp(-i*(phase21)); % Addition of phase1 aberration on ...
    config 1 aperture

%% FFT i.e. DFT on the aperture with aberration mask
Y = fft2(aperture_ab_21);
Y = abs(fftshift(Y)).^2;
figure(41)
imagesc(Y/max(max(Y)))
colormap(gray)
title('PSF: DFT phase 2 config 1');
%% Plotting the effect on Lena.csv
Y_dim = size(Y); % dimension of PSF image
Y_h = Y_dim(1);  % PSF image height
Y_w = Y_dim(2);  % PSF image width
% lenabuf = padarray(lena, [(Y_h/2 -ph/2) (Y_w/2 - pw/2)],0); % Resize object ...
    intensity distribution
lenabuf = imresize(lena, [Y_h Y_w]);
Y = conv2(lenabuf,Y,'same');
figure(49)
imagesc(Y/max(max(Y)))
colormap(gray)
title('Config 1 + Phase 2 + Lena');
%% Configuration 2 aperture generation
nxa = 640; % number of pixels in the x-axis
nya = 640; % number of pixels in the y-axis

x0a = nxa/2; % center of aperture
y0a = nya/2; % center of aperture
Ra = 128/2; % radius of pupil aperture

aperture_2 = zeros(nxa,nya); % aperture mask
countx = 0;
for z = (-nxa/2):1:(nxa/2)
    countx = countx +1;
    county = 0;
    for zz = (-nya/2):1:(nya/2)
        county = county +1 ;
        if (z)^2 +(zz)^2 < Ra^2
            aperture_2(countx,county) = 1; % Setting the value withing the radius to 1
        end
    end
end

% Inclusion of abberation phase term
phase22 = flipud(phase22);
aperture_ab_22 = aperture_2.*exp(-i*(phase22)); % Addition of phase1 aberration on ...
    config 2 aperture
%% FFT i.e. DFT on the aperture with aberration mask
Y = fft2(aperture_ab_22);
Y = abs(fftshift(Y)).^2;
```

```
figure(42)
imagesc(Y/max(max(Y)))
colormap(gray)
title('PSF: DFT phase 2 config 2');

%% Plotting the effect on Lena.csv
Y_dim = size(Y); % dimension of PSF image
Y_h = Y_dim(1);  % PSF image height
Y_w = Y_dim(2);  % PSF image width
% lenabuf = padarray(lena, [(Y_h/2 -ph/2) (Y_w/2 - pw/2)],0); % Resize object ...
    intensity distribution
lenabuf = imresize(lena, [Y_h Y_w]);
Y = conv2(lenabuf,Y,'same');
figure(50)
imagesc(Y/max(max(Y)))
colormap(gray)
title('Config 1 + Phase 2 + Lena');
```

References

[1] Goodman, J. W. *Introduction to Fourier Optics 2nd Edition*. McGraw-Hill, 1988.

[2] Michel Verhaegen, Paulo Pozzi,Oleg Soloviev, Gleb Vdovin, Dean Wilding. *Control for High Resolution Imaging*. Lecture notes for the course SC42030 TU Delft, 2017.

[3] Rodger E. Zeimer, William H. Tranter and D. Ronald Fannin, Compiled by Max Mulder and Christian Tiberius. *Instrumentation and Signals: Continuous and Discrete*. Pearson Custom Published for TU Delft., 2011.

[4] Wyant, J. C. Basic Wavefront Aberration Theory for Optical Metrology. Applied optics and optical engineering vol xi. University of Arizona, Tucson, Arizona, https://wp.optics.arizona.edu/jcwyant/wp-content/uploads/sites/13/2016/08/03-BasicAberrations_and_Optical_Testing.pdf.